

ساختمان داده ها و الگوریتم ها

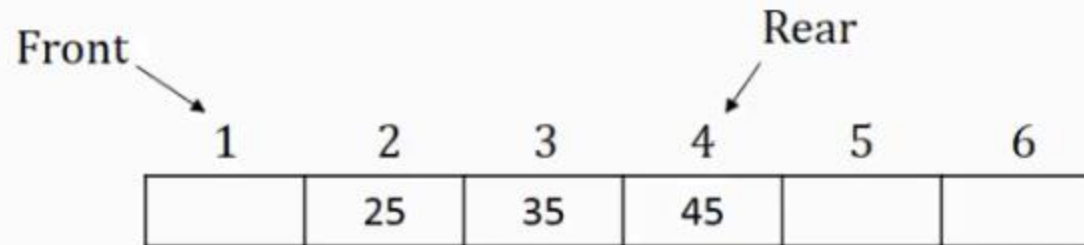
فصل چہارم

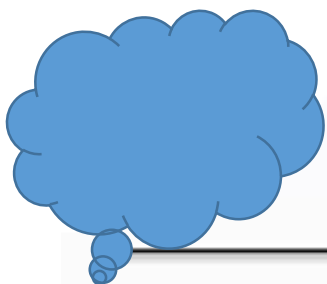
صف (Queue)

تعریف صف

به ساختار داده ای که در آن عمل درج از یک طرف (انتها=Rear) و عمل حذف از طرف دیگر (ابتدا=Front) انجام شود، صف گفته می شود. (First In First Out)

★ اشاره گر front به یکی کمتر از موقعیت اولین عنصر صف اشاره می کند و اشاره گر rear به موقعیت آخرین عنصر صف اشاره می کند.





پیاده سازی صف

برای ذخیره سازی ساختار یک صف می توان یکی از ۲ ساختار زیر را در نظر گرفت:

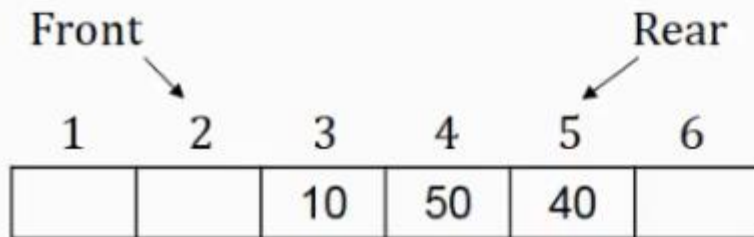
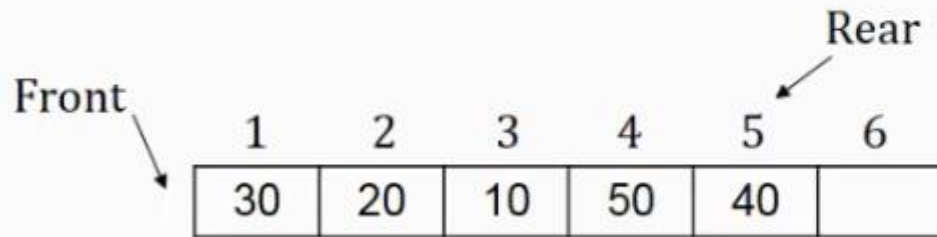
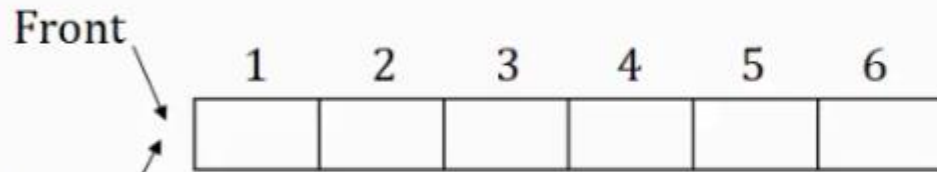
۱- آرایه: که ساختاری ایستا و محدود است.

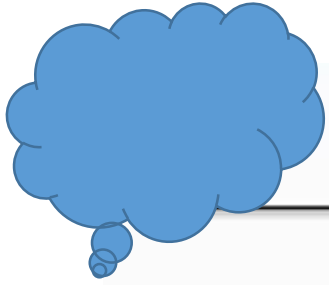
۲- لیست پیوندی: که ساختاری پویا و متناسب با داده هاست.

صف خطی

صف خطی آرایه ای n عنصریست که عناصر از ابتدای صف (Front) حذف شده و به انتهای صف (Rear) اضافه

می شوند.





صف خطی

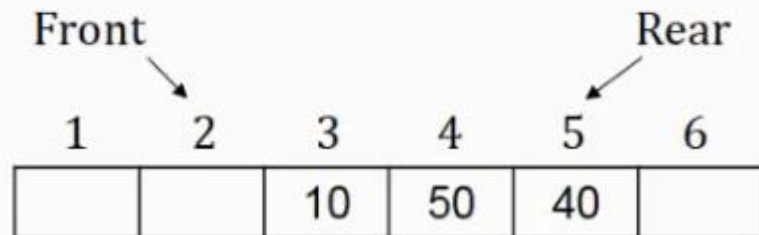
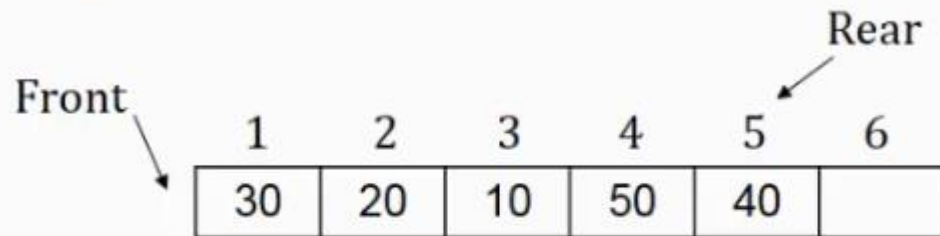
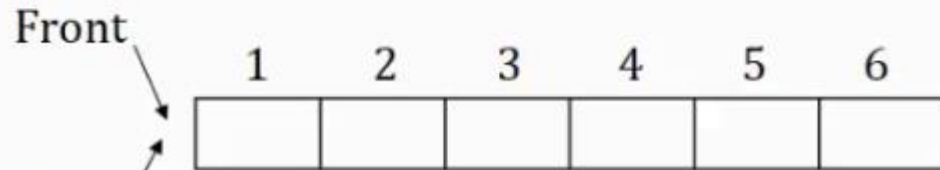
با فرض اینکه Front به خانه قبل از عنصر اول صف اشاره کند و Rear به عنصر آخر صف اشاره کند:

$Rear = Front = 0$	وضعیت اولیه صف
$Rear = Front$	خالی بودن صف
$Rear = n$	پر بودن صف

صف خطی

صف خطی آرایه ای n عنصریست که عناصر از ابتدای صف (Front) حذف شده و به انتهای صف (Rear) اضافه

می شوند.



درج در صف خطی


```
void addq ( type item ){  
  
    if ( rear == n )  
  
        cout << "Queue is FULL";  
  
    else  
  
        rear+=1;  
  
        Q[rear]=item;  
  
}
```

★ نمایش جمله "Queue is FULL" نشان دهنده

پر بودن صف نیست، زیرا ممکن است $\text{rear} =$

MaxQueueSize باشد و $\text{front} > 0$.

حذف از صف خطی



```
type deleteq ( ){
    if ( front == rear ){
        cout << "Queue is EMPTY";
        return 0;}
    else{
        front+=1;
        item=Q[front];
        return item;}
}
```



مشکل صف خطی

حرکت تدریجی عناصر به سمت انتهای صف و عدم امکان استفاده از خانه های ابتدای صف که در اثر حذف خالی شده اند.

راه حل:

(۱) شیفت خانه های انتهای صف به سمت ابتدای صف (در صورتی که تعداد خانه های صف زیاد باشد هزینه زیادی

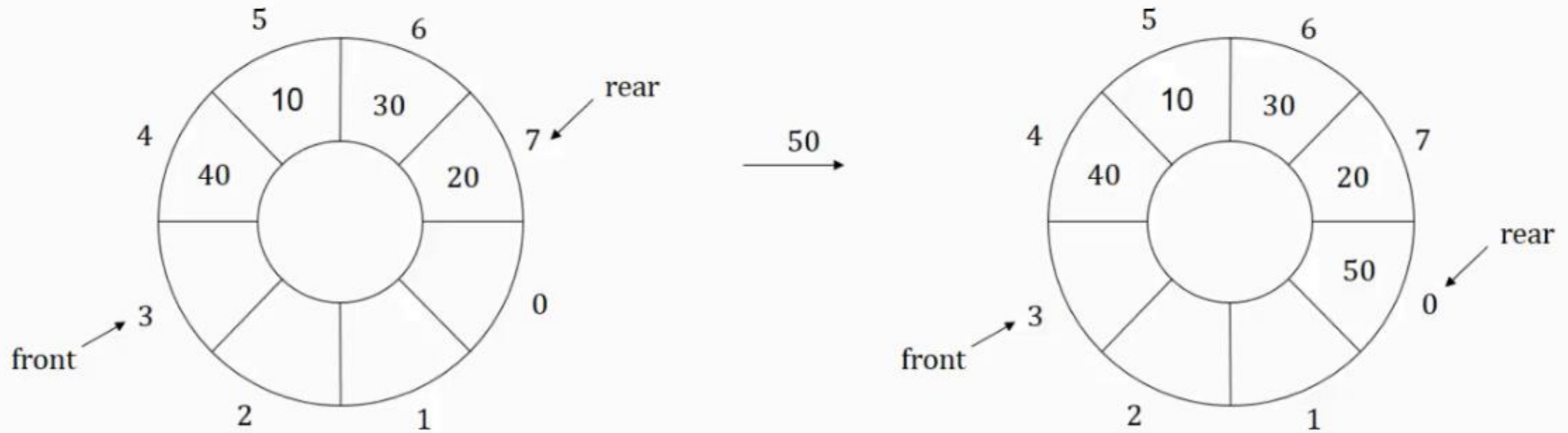
خواهد داشت.)

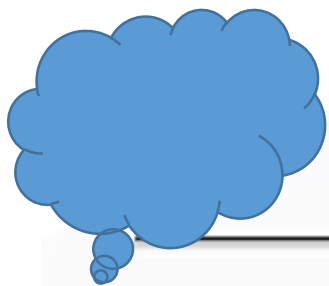
(۲) استفاده از صف حلقوی: یعنی هنگامی که به انتهای صف رسیدیم عمل درج را مجدداً از ابتدای صف انجام می

دهیم.

صف حلقوی

آرایه ای n عنصری را می توان به صورت یک صف حلقوی در نظر گرفت به طوری که زمانی که $rear$ برابر $n-1$ می شود عنصر بعدی در خانه 0 قرار گیرد.





صف حلقوی

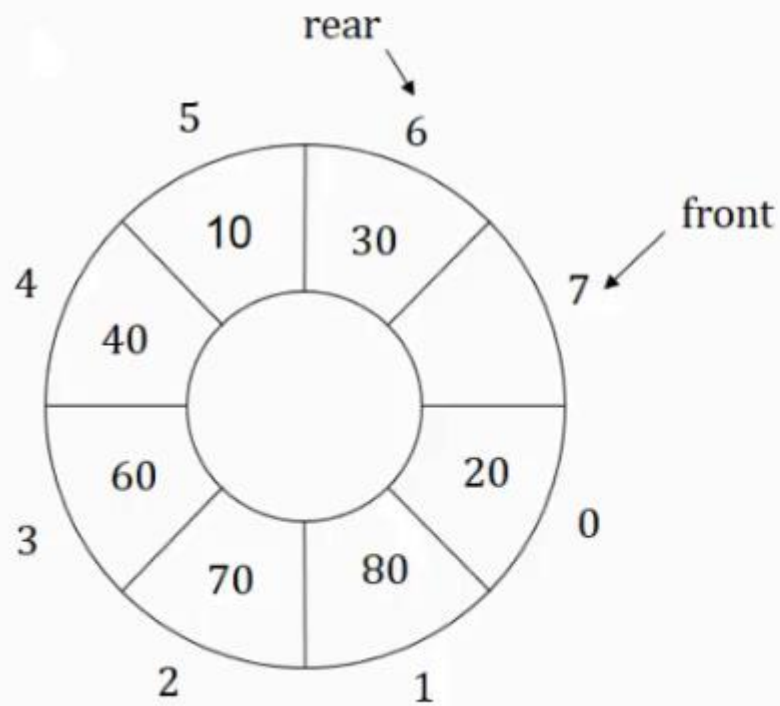
با فرض اینکه Front به خانه قبل از عنصر اول صف اشاره کند و Rear به عنصر آخر صف اشاره کند:

$Rear = Front = 0$	وضعیت اولیه صف
$Rear = Front$	خالی بودن صف
$Front = (Rear + 1) \% n$	پر بودن صف

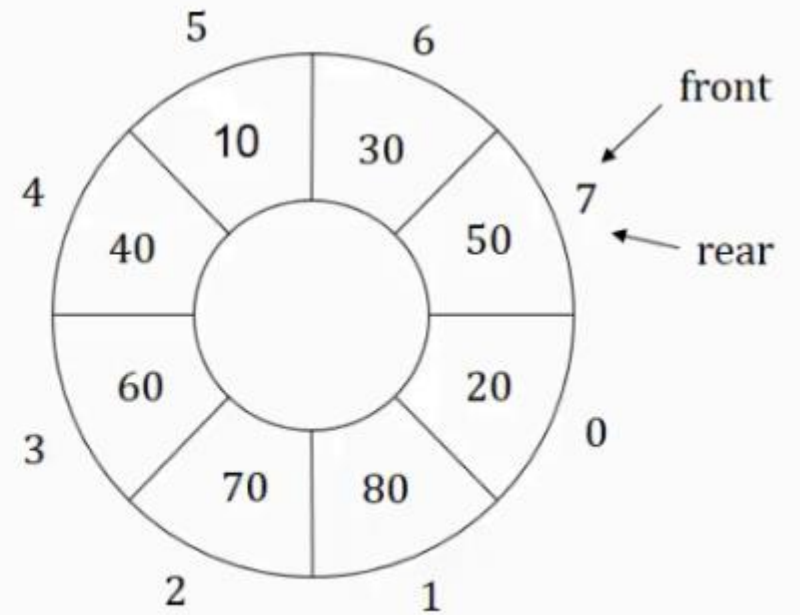
صف حلقوی

★ در صف حلقوی با ظرفیت n همیشه باشد یک خانه خالی باشد، به دلیل این که بتوان وضعیت پر یا خالی بودن

صف حلقوی را تشخیص داد



50



درج در صف حلقوی

```
void addq ( type item ){
int t;
t=(rear+1)%n;
if ( t == front )
    cout <<“Queue is FULL”;
else
    rear=t;
    Q[rear]=item;
}
```

حذف از صف حلقوی

```
type deleteq ( ){  
    if ( front == rear ){  
        cout << "Queue is EMPTY";  
        return 0;}  
    else{  
        front=(front+1)%n;  
        item=Q[front];  
        return item;}  
}
```